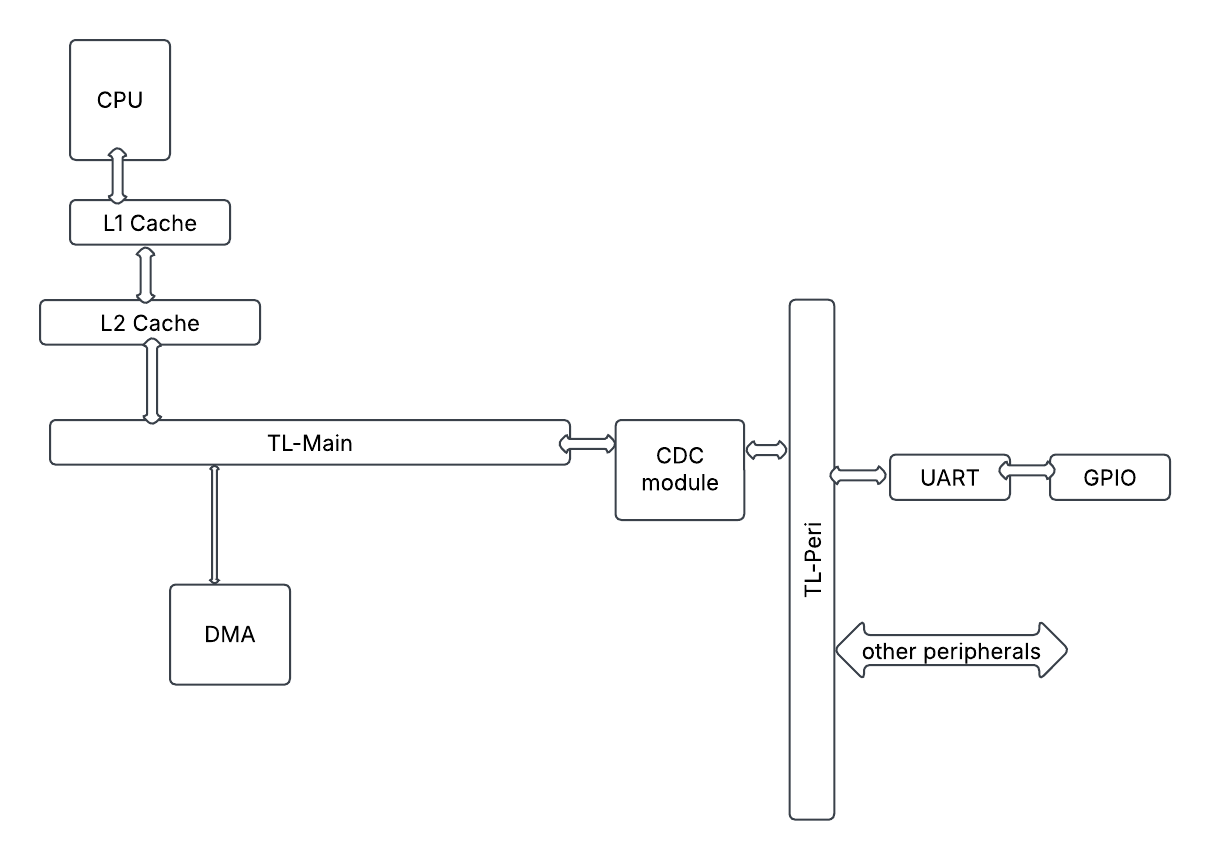
Direct Memory Access Architecture and Controller



CPU configures DMA registers with source address, destination address, and the size of data to be transferred from either memory to peripheral or vice-versa, and sets the ‘go’ register. DMA looks for the set value of ‘go’ to start the process of reading from the source address and writing to the destination. As of now, DMA communicates with the L2 cache on the high speed side, and GIPO through UART on the peripheral side. For example, if CPU configured DMA registers to read from memory and write to a peripheral, this is how the data transfer works. Once the register ‘go’ is set, DMA places a read request to the TL-main and receives the data value from the memory through D-channel response. As the data is received, DMA sends a write request on a-channel to the TL-Main with the address details of the peripheral and the data value, TL-Main decodes the address is intended for a peripheral and routes it to TL-Peri through the CDC module. As DMA receives a positive response, it updates the ‘done’ register, and raises an interrupt signal to the CPU, and waits on the next transfer request from CPU. CPU upon receiving interrupt, proceeds to check the ‘done’/’error’ registers and proceeds to raise another request or act in case of an error message. Also, after CPU configures DMA for a transfer request and sets ‘go’ register, DMA requests for TL-Main bus access and CPU grants the request and relinquishes control of the bus.

